Call Admission Control in Wireless LAN

Boris Bellalta¹ and Michela Meo²
(1) UPF, Spain
(2) Politecnico di Torino, Italy
Corresponding author: boris.bellalta@upf.edu

1 BibTex

@ARTICLE{Bellalta80211AdControlModel,
  AUTHOR = {Boris Bellalta and Michela Meo},
  JOURNAL = {Auerbach Publications, Taylor and Francis Group},
  YEAR = {2006},
}


Contents

1 BibTex i

2 Introduction 1

3 Related Work 2

4 A HotSpot Wireless Scenario 5
  4.1 MAC protocol description ........................................ 6
  4.2 System parameters ............................................... 7
  4.3 Frame Durations ................................................ 7
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>A Model of the IEEE 802.11 MAC layer</td>
<td>8</td>
</tr>
<tr>
<td>5.1</td>
<td>A mobile node</td>
<td>8</td>
</tr>
<tr>
<td>5.2</td>
<td>The MAC protocol</td>
<td>9</td>
</tr>
<tr>
<td>5.3</td>
<td>Model validation</td>
<td>11</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Homogeneous traffic flows</td>
<td>11</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Heterogeneous traffic flows</td>
<td>12</td>
</tr>
<tr>
<td>5.4</td>
<td>Model applications</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>User-level performance in Hot-Spot WLANs</td>
<td>13</td>
</tr>
<tr>
<td>6.1</td>
<td>A general view of TCP performance</td>
<td>14</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Downlink TCP flows</td>
<td>15</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Uplink TCP flows</td>
<td>17</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Simultaneous Downlink and Uplink TCP flows</td>
<td>18</td>
</tr>
<tr>
<td>6.2</td>
<td>Voice over IP</td>
<td>19</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Protocol Overheads</td>
<td>20</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Unfairness</td>
<td>20</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Interaction with TCP flows</td>
<td>21</td>
</tr>
<tr>
<td>7</td>
<td>Call Admission Control</td>
<td>22</td>
</tr>
<tr>
<td>7.1</td>
<td>CAC Architecture</td>
<td>22</td>
</tr>
<tr>
<td>7.1.1</td>
<td>A dual queue scheme at the Access Point</td>
<td>22</td>
</tr>
<tr>
<td>7.1.2</td>
<td>Uplink differentiation via different values of $CW_{\text{min}}$</td>
<td>23</td>
</tr>
<tr>
<td>7.2</td>
<td>Proposed algorithm</td>
<td>23</td>
</tr>
<tr>
<td>7.3</td>
<td>Performance Evaluation</td>
<td>24</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Model of a cell</td>
<td>24</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Parameters</td>
<td>25</td>
</tr>
<tr>
<td>7.3.3</td>
<td>Considered metrics</td>
<td>26</td>
</tr>
<tr>
<td>7.4</td>
<td>Numerical Results</td>
<td>27</td>
</tr>
<tr>
<td>8</td>
<td>Conclusions</td>
<td>28</td>
</tr>
</tbody>
</table>
List of Figures

1. Example of the basic access mechanism ................................................. 38
2. Sketch of the considered network ......................................................... 38
3. Homogeneous case - Aggregate throughput for $S_1$, $S_2$ and $E_1$ traffic profiles .............................. 39
4. Heterogeneous case - Aggregate throughput for $S_1$ and $S_2$ nodes in scenario 1 ......................... 39
5. Heterogeneous case - Aggregate throughput for $S_1$, $S_2$ and $E_1$ nodes in scenario 2 .......... 40
6. Discrete Markov Chain describing the evolution of the number of backlogged nodes ........ 40
7. VoIP - Throughput (AP and mobile nodes) for two voice codecs (G.711 and G.729) ............. 41
8. $BP_s$ - Blocking probability VoIP calls .................................................. 41
9. $BP_{e,u}$ - Blocking probability for elastic uplink flows ................................ 42
10. $BP_{e,d}$ - Blocking probability for elastic downlink flows .............................. 42
11. $ES_{e,u}$ - Average elastic throughput uplink .......................................... 43
12. $ES_{e,d}$ - Average elastic throughput downlink ...................................... 43
List of Tables

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System parameters of the IEEE 802.11b specification [1]</td>
</tr>
<tr>
<td>2</td>
<td>Protocol overheads from upper layers</td>
</tr>
<tr>
<td>3</td>
<td>Model validation. Traffic profiles</td>
</tr>
<tr>
<td>4</td>
<td>S1 - Homogeneous case. Model validation for several performance parameters</td>
</tr>
<tr>
<td>5</td>
<td>S2 - Homogeneous case. Model validation for several performance parameters</td>
</tr>
<tr>
<td>6</td>
<td>S1, S2 - Queue utilization, scenario 1</td>
</tr>
<tr>
<td>7</td>
<td>Conditional collision probability, expected number of slots of each backoff instance and channel probabilities (S1), scenario 2</td>
</tr>
<tr>
<td>8</td>
<td>S1, S2 - Mutual interactions among flows, scenario 1</td>
</tr>
<tr>
<td>9</td>
<td>$S_{tcp}^{e,d}$ and $S_{tcp}^{e,u}$ - Aggregate throughput for persistent TCP flows (Mbps) - $L_{tcp} = 1500$ bytes (including the TCP header)</td>
</tr>
<tr>
<td>10</td>
<td>$S_{tcp}^{e,d}$ - Comparison of TCP downlink models (Mbps) - $L_{tcp} = 1500$ bytes (including the TCP header)</td>
</tr>
<tr>
<td>11</td>
<td>$S_{tcp}^{e,u}$ - Comparison of TCP uplink performance models (Mbps) - $L_{tcp} = 1500$ bytes (including the TCP header)</td>
</tr>
<tr>
<td>12</td>
<td>$S_{tcp}^{e,d},S_{tcp}^{e,u}$ - Comparison of simultaneous TCP downlink/uplink flows (Mbps) - $L_{tcp} = 1500$ bytes (including the TCP header)</td>
</tr>
<tr>
<td>13</td>
<td>Typical values of most used codecs in VoIP</td>
</tr>
<tr>
<td>14</td>
<td>VoIP efficiency over WLAN</td>
</tr>
<tr>
<td>15</td>
<td>Bandwidth share between uplink and downlink VoIP flows</td>
</tr>
<tr>
<td>16</td>
<td>Downlink throughput for VoIP calls in presence of TCP flows (Mbps)</td>
</tr>
</tbody>
</table>
2 Introduction

The increasing popularity of Wireless Local Area Networks (WLANs) based on the IEEE 802.11 technology, due to the ease in installing access facilities and to the affordable price of the equipments, is pushing operators to deploy WiFi WLANs as access networks to their services. However, the limitations of this technology, such as the still limited radio resources, the poor channel quality depending on relative position of Mobile Nodes (MN), the interference from hidden terminals, and the anomaly observed when MNs transmit at different speeds, make it difficult to cope with the need to provide a variety of services with different characteristics in terms of QoS requirements. Indeed, current WLAN networks provide best-effort services without any QoS guarantees. Due to the higher bandwidth provided, multimedia services such video and voice streaming can perform well under low load conditions but, when traffic intensity increases, the delay and bandwidth of streaming flows are severely affected, thus degrading the received quality of service.

Solutions are therefore needed for service differentiation at the access and for providing QoS guarantees. In particular, services can be distinguished in two classes based on the mechanisms employed at the transport layer: i) elastic flows, usually adopted to deliver data services such as file transfer applications, correspond to traffic carried by TCP and TCP-like protocols, that adapt the traffic generation rate to the network working conditions, attempting in this way to reduce network congestion; ii) streaming flows, adopted by multi-media applications, tend to generate traffic unaware and independently of the network conditions.

Based on this differentiation, some possible solutions for guaranteeing acceptable QoS levels consist in performing call admission control (CAC). Flows can be accepted as far as the total traffic is below a given threshold. By properly choosing the threshold, QoS levels for accepted flows can be reasonably guaranteed. Moreover, service differentiation can be achieved by applying the scheme to streaming or elastic flows, or both.

In this chapter, we first propose a model of the IEEE 802.11 MAC layer and investigate the performance perceived by streaming and elastic flows in a Hot-Spot scenario. We then use the model as the core of a CAC scheme. Indeed, the scheme is based on a bi-dimensional Markov Chain describing the dynamics of flow arrivals and completions for each class of traffic. The transition rates in the chain are set according to the MAC layer model.
3 Related Work

The IEEE 802.11 MAC and the PHY layer specifications for the 802.11b standard in the 2.4 GHz band are explained in [1]. For other PHY specifications refer the amendments [2] (802.11a) or [3] (802.11g). These posterior PHY specifications allow the system to achieve higher data rates by improving the modulation and coding techniques, the MAC specification remaining unchanged. The current version of the MAC protocol does not implement any QoS mechanism, which is solved with the IEEE 802.11e standard [4].

Since the first specifications of the IEEE 802.11 appeared, a large amount of research has been done to analytically model the IEEE 802.11 MAC protocol. We remark the seminal papers of Bianchi [5] and Tay et al. [6], referred in most of posterior works. In these two papers, the authors address the performance analysis of the MAC protocol assuming a finite number of saturated sources (i.e., sources always have a packet ready to be transmitted) which compete for the use of the shared channel. In both papers, the authors find the value of the MAC parameters which maximize the network throughput. The decoupling assumption (i.e., nodes attempt to transmit to the channel independently from each other) is applied to simplify the protocol analysis which can be solved through a fixed point procedure. The presented results can be used to understand how the MAC protocol performs or to obtain measures such as the maximum throughput given a number of nodes in the system. These papers are well complemented by the delay analysis of the MAC protocol presented in [7], where Carvalho et al. obtain the first and second moment of the service time, still in saturation conditions. It is remarkable that the authors find closed form expressions by linearizing the fixed point equations presented in the previous mentioned papers.

Models of the saturated system are unable to provide more detailed information about the behavior of the network (actual load, network utilization, etc.) or of each node individually (transmission attempt rate, collision probability, queuing delays, user throughput, etc.) which are of crucial importance to compute the grade of service the traffic flows receive from the network. Furthermore, protocol enhancements by means of dynamic tuning of backoff parameters or other MAC parameters have marginal effects when the network is un-saturated and, therefore, the real gain is also marginal. In recent years, several papers addressed the modeling task of the IEEE 802.11 MAC performance in non-saturation conditions. Two groups of papers can be found, based on a
Markovian stochastic analysis, which are extensions of the model presented by Bianchi [5]. In [8],
the authors introduce new states in the Markov chain that describe the backoff algorithm, model-
ing the time in which the mobile nodes are empty (i.e., has no packet ready to be transmitted).
Another group of papers is based on the observation that the attempt rate of a node is a regen-
erative process and it can be computed by using the renewal-reward theorem. This approach is
used in [9], [10] and [11]. Surprisingly, the decoupling assumption works well also in non-saturated
conditions and results obtained by both type of models are accurate. One of the benefits of the
un-saturated system models is that they allow to analyze the system performance when nodes have
different (heterogeneous) traffic profiles. For example in [8] the authors analyze the system under
the presence of streaming and elastic flows, respectively carried by UDP and TCP transport layer
protocols.

Today, WLAN networks are basically used to access Internet and download web pages or other
types of information. Therefore, the most part of data is carried by the TCP protocol from the
Internet to the end user (i.e., most of the traffic flows in the downlink direction). The first work
that analytically addresses the performance of a WLAN with TCP traffic is [12]. The authors
consider both the data traffic in the downlink and the feedback traffic in the uplink due to TCP
ACKs, modeling the access point and the mobile nodes as saturated sources. In order to catch
the effect that the number of backlogged nodes is not constant and depends on the number of
TCP connections, the authors propose the use of a discrete-time Markov chain to obtain the
probability that \( n \) nodes have an ACK ready to be transmitted, and thus compete with the AP
to transmit on the channel. The average system throughput is obtained by means of a time-
scale decomposition. A similar approach is used in [13], including the effect of delayed ACK
techniques and the presence of short-lived TCP connections. More recently, in [14] this problem
is also considered under heterogeneous radio conditions. Similar results are obtained in the three
papers. Finally, in [15], the authors analyze the presence of downlink/uplink and bidirectional TCP
flows but under the assumption that the TCP advertisement window is equal to one.

The introduction of VoIP services over WLAN networks is a going to further increase the use
of WLANs, and it is a hot topic in the research community. In papers like [16], [11], [17] or [18],
a capacity analysis is presented. Similar conclusions are presented in all of them, remarking that
the access point is the bottleneck of the network. To avoid this problem and increase the capacity,
several solutions can be considered from the IEEE 802.11e specifications [4], for example by using the TXOP (Transmission Opportunity) option [17].

Therefore, motivated by the expected grow of multimedia traffic in future WLAN, a lot of work has been done in resource allocation strategies and admission control under the goal to provide mechanisms which assure differentiated services. To differentiate among traffic flows, three main strategies are used in the literature, based on the specifications of [4]: i) different DIFS values for each QoS level, ii) setting a different $CW_{min}$ value to each flow [19, 20] and iii) using the Transmission Opportunity option (TXOP) [17, 21]. The first one assures rigid flow differentiation but second and third allow a more fine tuning of the service received by each flow. In these schemes, admission control is mandatory to prevent that the system remains always in a stable state, blocking if necessary new flows. Two groups of admission control schemes exists: i) model based [22], [20], [23] which estimate the future system status using mathematical models of the system and ii) measure based [24] which predict the future system status from current measures. Anyway, this is a soft classification since the major part of CAC schemes uses both models and measure information to achieve their goal.

One of the main difficulties of admission control schemes is to predict the future system state using actual system information as the non-linear behavior of the IEEE 802.11 MAC protocol parameters (conditional collision probability, transmission probability, mobile node queue utilization, etc.) with the number of flows and their traffic characteristics. For example, in Pong et al. [22], in order to compute the effect of the introduction of a new flow, the admission control assigns to the new flow the characteristics obtained from a similar throughput flow.

Banchs et al. propose in [20] an admission control scheme and a parameter tuning algorithm for the $CW_{min}$ parameter assuming that $CW_{max} = CW_{min}$, or that the parameter $m = 0$. The rational behind the assumption is that, if the admission control provides the optimal $CW_{min}$, the fact that $m > 0$ can tend to suboptimal situations. The CAC is also based on the model of [5]. For each new request, the CAC estimates the collision probability by assuming that all nodes are saturated (thus, the conditional collision probability is the same for all flows) and computes the system achievable throughput. Under the assumption that the transmission probability is proportional to the throughput requested by each flow, the CAC computes the individual achievable throughput. It is remarkable the derivation of optimal $CW_{min}$ values which allows the system to maximize the
number of active flows.

In [24] the HARMONICA architecture for admission control and parameter tuning is presented. It uses LQI (Link Quality Indicator) to catch metrics such as packet dropping, link end-to-end delay and throughput which are used to decide if a new flow can be admitted and the optimum parameters to be assigned to each flow in order to maximize the network utilization.

Finally, in [23] an admission control scheme for both streaming and elastic flows is presented. The admission control admits or rejects streaming flows and adjusts the transmitting rate of elastic flows to avoid that they interfere with streaming flows. The authors observe how the business ratio (fraction of time the channel is not empty) is practically equal to the channel utilization (fraction of time the channel is transmitting successful frames), independently of the number of users. One of the conclusions of the paper is that, by maintaining the business ratio close to a certain threshold, the system throughput is maximized at the same time that delay and delay variation are minimized. The authors use the normalized throughput which is linear with the business ratio until the selected threshold is reached (which is about 0.9 of the channel utilization for both RTS/CTS and BA access schemes). For the rate control operation, it is assumed that a traffic shaping procedure implemented at the nodes and the access point regulates the TCP traffic offered to the network.

4 A HotSpot Wireless Scenario

A wireless Cell (or hot-spot) is the coverage area provided by a single access point (in [1] it is referred as a Basic Service Set, BSS). The coverage area is the geographical area where both the access point (AP) and the mobile stations can communicate using the radio channel with an acceptable minimum quality; this quality can be measured in terms of SNR and other derived metrics such as the Frame Error Ratio (FER). An Extended Service Set (ESS) contains multiple access points and their coverage areas. All or part of these coverage areas can overlap, so that a mobile station can select the access point to use; we call these areas reassociation or handoff areas.

Typical scenarios with this configuration are found in public areas (like cafeterias, parks, airports) where users can access the Internet from their notebooks or PDAs; company buildings where workers use WLAN networks to communicate through the email service, message applications or voice over IP; individual users at their homes, etc. In all these scenarios, the WLAN technology
provides a certain grade of mobility and a broadband access to Internet at very low cost.

In this work, we consider a single BSS with an AP providing access to a fixed network to \( n \) mobile nodes. Each node has a traffic profile specifying its basic configuration parameters, i.e., bandwidth, packet arrival rate, expected frame length, etc. The mobile nodes and the access point use the DCF (Distributed Coordination function) of the IEEE 802.11 MAC and the DSSS PHY specifications in the 2.4 \( GHz \) band[1].

### 4.1 MAC protocol description

The IEEE 802.11 medium access control is based on a distributed CSMA/CA protocol [1]. According to the basic access (BA) mechanism, when a node has no packets to transmit and receives a packet from the network layer, the node starts to sense the channel to determine its state, that can be either busy or free. If the channel is detected busy, the node waits until the channel is released. When the channel is detected free for a period of time larger than the DIFS (Distributed Inter-Frame Spacing) duration, a new backoff instance is generated. A backoff instance consists on a counter set to a random value each time it is generated. The random value is picked from a uniform distribution in the range \( CW(k) = \left[ 0, \min \left( 2^k CW_{\text{min}} - 1, 2^m CW_{\text{min}} - 1 \right) \right] \), where \( k \) is the current attempt to transmit the packet, \( CW_{\text{min}} \) is the minimum size of the contention window, and \( m \) defines the maximum size of the window. For each packet to be transmitted, \( k \) is initially set to 0 and it is increased by one at each failed transmission until a maximum number of retransmissions, called Retry Limit, is reached, and the packet is dropped. The counter is decreased by one for each time-slot \( \sigma \) in which the channel is sensed free, and, when the countdown reaches zero, the node starts the packet transmission on the channel. If during the backoff countdown the channel is sensed busy, the backoff is suspended until the channel is detected free again.

A collision occurs if two nodes transmit at the same time, i.e., the backoff instances from both nodes reach 0 at the same time. After the data packet is transmitted by the sender, the receiver waits for a SIFS (Short Inter-Frame Spacing) time and sends a MAC layer ACK to acknowledge the correct reception of the data packet. In the case the sender does not receive the ACK frame, it starts the retransmission procedure. After discarding or successfully transmitting a packet, if more packets are ready to be transmitted, the node starts the transmission procedure again. Otherwise, it waits for a new packet from the network layer. In Figure 1 we plot an example of the basic access
(BA) mechanism with three mobile stations contending to transmit a packet.

Alternative to the BA mechanism, nodes can employ a RTS/CTS protocol to access the channel, so as to reduce the hidden terminal effect.

4.2 System parameters

The system parameters are reported in Table 1, the overhead introduced by upper layers are listed in Table 2. We assume ideal channel conditions, i.e., no packet is lost due to channel errors or the hidden terminal phenomenon. In Figure 2 a sketch of the considered network is presented. The fixed network is modeled by a simple 100 Mbps full duplex link with a propagation delay of 2 ms in both directions. This link is used to interconnect a fixed node (server) where one end-point of the traffic flows resides. The other end-points are in the mobile nodes, which are linked to the server through the access point.

4.3 Frame Durations

When a node transmits a frame, two possible events can occur: a collision or a successful transmission. The duration of both events depends on the employed access mechanism. The successful transmission duration for the BA and the RTS/CTS mechanisms, are, respectively, given by:

\[
T_{s}^{ba} = \frac{PHY_{H}}{R_{basic}} + \frac{MAC + L_{data} + MAC_{FCS}}{R_{data}} + SIFS + \frac{PHY_{H}}{R_{basic}} + \frac{L_{ACK}}{R_{basic}} + DIFS
\] (1)

\[
T_{s}^{rts} = O_{rts} + \frac{PHY_{H}}{R_{basic}} + \frac{MAC + L_{data} + MAC_{FCS}}{R_{data}} + SIFS + \frac{PHY_{H}}{R_{basic}} + \frac{L_{ACK}}{R_{basic}} + DIFS
\] (2)

with

\[
O_{rts} = \frac{PHY_{H}}{R_{basic}} + \frac{RTS}{R_{basic}} + SIFS + \frac{PHY_{H}}{R_{basic}} + \frac{CTS}{R_{basic}} + SIFS
\] (3)

\[
PHY_{H} = PLCP \text{ preamble} + PLCP \text{ header}
\] (4)
For the basic access mechanism, the duration of a collision is equal to the maximum successful transmission duration of the colliding frames, but for the RTS/CTS mechanism the duration of a collision is constant and equal to

\[ T_{c}^{rts} = \frac{PHY}{R_{basic}} + \frac{RTS}{R_{basic}} + EIFS \]  

(5)

The additional overhead of the RTS/CTS access compensates with a low collision duration.

5 A Model of the IEEE 802.11 MAC layer

In this section, an user-centric model of the DCF function of the 802.11 MAC layer is presented. We approximate each mobile node by a finite length queue with network-dependent service time.

5.1 A mobile node

Packets with average length \( L_i \) arrive to node \( i \) with rate \( \lambda_i \). Both the time between packet arrivals and the service time are assumed to be exponentially distributed. Therefore, a mobile node (the AP included) is modeled by an \( M/M/1/Q_i \) queue with \( Q_i \) as the queue length measured in packets.

The offered traffic load to the MAC layer and the queue utilization for node \( i \) are \( \nu_i = \lambda_iX_i \) and \( \rho_i = \lambda_i(1 - P_{b,i})X_i \) respectively, where \( X_i \) and \( P_{b,i} \) are the mean service time and the packet blocking probability. The node throughput is \( S_i = \rho_iL_i/X_i \).

By modeling each mobile node using an \( M/M/1/Q_i \) queue we can obtain simple expressions to measure the quality of the service observed by a node in terms of blocking probability, average queue length and average transmission delay (including the service time).

\[ P_{b,i} = \frac{\nu_i^Q_i}{\sum_{j=0}^Q_i \nu_i^j} \quad EQ_i = \frac{\sum_{j=0}^Q_i j\nu_i^j}{\sum_{j=0}^Q_i \nu_i^j} \quad ED_i = \frac{EQ_i}{\lambda_i(1 - P_{b,i})} \]  

(6)

Finally, the probability to loose a packet is the probability that the packet is discarded at the queue entrance due to overflow or dropped at the MAC layer because the number of retransmissions have exceeded the retry limit, \( R_i \). Then, a packet loss occur with probability

\[ P_{L,i} = P_{b,i} + (1 - P_{b,i})P_{d,i} \]  

(7)
where \( P_{d,i} \) is the probability that a packet is dropped at the MAC layer.

### 5.2 The MAC protocol

A node with a packet ready to be transmitted starts a backoff instance. Letting \( EB_i \) be the average number of slots selected by node \( i \) at each transmission attempt, the steady state probability that the node transmits in a random slot given that a packet is ready in its transmission queue can be computed from

\[
\tau_i = \frac{E[Pr(Q_i(t) > 0)]}{EB_i + 1} = \frac{\rho_i}{EB_i + 1}
\]

(8)

Node \( i \) transmission collides if any other node also transmits in the same slot. Then, the conditional collision probability for node \( i \) is

\[
p_i = 1 - \prod_{j \neq i} (1 - \tau_j)
\]

(9)

In order to compute \( EB_i \), two different approaches are found in the literature: a stochastic (Markovian) approach [5] and an average analysis [6]. Expressions found in both papers are different but numerically equal. For simplicity, we choose to use the expression of [6], then \( EB_i \) is computed as

\[
EB_i = \frac{1 - p_i - p_i(2p_i)^{\rho_i} C W_{\text{min}}}{1 - 2p_i} - \frac{1}{2}
\]

(10)

The effect of the Retry Limit \( R_i \) is considered in [25]. However, for operative values of \( p_i < 0.4 \), the effect of \( R_i \) on the average backoff time at each attempt is almost negligible. Using the conditional collision probability, the dropping probability at the MAC layer is given by the probability that a packet collides \( R_i \) times, \( P_{d,i} = p_i^{R_i} \).

The service time, i.e., the time interval from the instant in which a packet enters in service until it is completely transmitted or discarded, is given by,

\[
X_i = (M - 1) \left( EB_i \alpha_i + ET_{c,i}^{bs} |rt_s \right) + EB_i \alpha_i + T_{s,i}
\]

(11)

where \( M \) is the average number of transmissions, \( \alpha_i \) is the average slot duration and \( ET_{c,i} \) is
the average duration of a collision of node \(i\). We approximate the value of \(ET_{c,i}\) by,

\[
\begin{align*}
ET_{c,i}^{ba} &\approx \frac{\sum_{j \neq i} \tau_j \max(T_{s,j}, T_{s,i})}{\sum_{j \neq i} \tau_j} \\
ET_{c,i}^{rts} & = T_{c}^{rts}
\end{align*}
\]

where we neglect the fact that more than two packets collide simultaneously. Note that if the RTS/CTS access scheme is used, \(ET_{c,i}^{rts}\) is constant and equal for all nodes. The average number of transmissions that a packet undergoes is computed under the decoupling assumption as,

\[
M = \frac{1 - p_i R_i + 1}{1 - p_i}
\]

A node freezes its backoff counter every time the channel is sensed busy and releases it after the channel is sensed free for a DIFS period. Therefore, the time between two backoff counter decrements is a random variable which depends on the behavior of the other nodes. By letting \(\alpha_i\) be the average time between two backoff counter decrements, or equivalently, the average slot duration, we have

\[
\alpha_i = p_{e,i} \sigma + p_{s,i}(ET_{s,i}^{ba||rts,*} + \sigma) + p_{c,i}(ET_{c,i}^{ba||rts,*} + \sigma)
\]

where \(ET_{s,i}^{ba||rts,*}\) and \(ET_{c,i}^{ba||rts,*}\) are the average durations of an observed successful transmission or a collision for node \(i\) when it is performing a backoff instance. To compute \(ET_{c,i}^{ba||rts,*}\) we consider that the probability that more than two stations collide can be neglected, then

\[
\begin{align*}
ET_{c,i}^{ba,*} &\approx \frac{\sum_{j \neq i} \sum_{k > j, k \neq i} \max(T_{s,j}, T_{s,k})(\tau_j \tau_k \prod_{u \neq (j,k,i)} (1-\tau_u))}{\sum_{j \neq i} \sum_{k > j, k \neq i} (\tau_j \tau_k \prod_{u \neq (j,k,i)} (1-\tau_u))} \\
ET_{c,i}^{rts,*} & = T_{c}^{rts}
\end{align*}
\]

and

\[
ET_{s,i}^{*} \approx \frac{\sum_{j \neq i} T_{s,j} \left(\tau_j \prod_{u \neq (i,j)} (1-\tau_u)\right)}{\sum_{j \neq i} \left(\tau_j \prod_{u \neq (i,j)} (1-\tau_u)\right)}
\]

The probabilities \(p_{e,i}\), \(p_{s,i}\) and \(p_{c,i}\) are related to the channel status in a given slot when a node is in backoff. \(p_{e,i}\) is the probability that a slot is observed empty, \(p_{s,i}\) the probability that in a slot a successful transmission occurs and \(p_{c,i}\) is the probability that a collision occurs. Note that at
the end of a successful transmission or a collision we add the duration of an empty slot, since the
backoff counter is only decreased after the channel is sensed empty for the full duration of a slot.
These channel probabilities can be computed as

\[ p_{e,i} = \prod_{j \neq i} (1 - \tau_j) \quad p_{s,i} = \sum_{z \neq i} \tau_z \prod_{j \neq z \neq i} (1 - \tau_j) \quad p_{c,i} = 1 - p_{e,i} - p_{s,i} \]  

(17)

Due to the dependence of previous expressions on the queue utilization of each node, \( \rho_i \), and
the fact that (8) and (9) form a set of non-linear equations, we have to use iterative numerical
techniques to solve the model.

5.3 Model validation

In order to validate the model and analyze the performance of the IEEE 802.11 MAC protocol, we
consider a single-hop scenario with three different types of flows whose characteristics are summa-
rized in Table 3. The network comprises \( n + 1 \) nodes including the access point, each node uses the
BA access scheme and carries a single traffic flow. We refer to streaming type 1 (streaming type
2) flows with \( S_1 \) (\( S_2 \)) and we use \( E_1 \) to refer to elastic flows.

Analytical results are compared against simulations performed using the ns2 package [26]. How-
ever, we have also built a detailed simulator of the IEEE 802.11 MAC protocol using the COST
(Component Oriented Simulation Toolkit) simulation package [27] and verified that it provides
equivalent results with respect to ns2 but allowing a higher flexibility to monitor the dynamics of
the MAC parameters.

5.3.1 Homogeneous traffic flows

A first validation is done considering that all nodes in the network have the same traffic profile (\( S_1, \)
\( S_2 \) or \( E_1 \)). Figure 3 shows predicted and simulated aggregate throughput against the number of
flows for the three traffic classes specified in Table 3. The analytical and simulation models bring
to very close results, showing the accuracy of the model.

As the elastic results are obtained from saturated nodes, equivalent results are obtained in
[5, 6]. For the unsaturated traffic flows, in Tables 4 and 5 we report the values of other parameters
such as the conditional collision probability \( p_i \), the queue utilization \( \rho_i \), the average queueing delay
and packet losses $P_{L,i}$. The model captures the non-linear dynamics of these parameters, specially the complex transition from the unsaturated to the saturated conditions. Note that under saturation conditions, as we expect, parameters like the conditional collision probability are equal and independent of the traffic load. Differences between the model and the simulations (the model is pessimistic) are mainly motivated by the assumption of a exponential packet length distribution in the model that is constant in simulation.

\section*{5.3.2 Heterogeneous traffic flows}

Having the model been evaluated in the homogeneous scenario, we now define an heterogeneous scenario. We chose a configuration where two types of streaming flows, $S_1$ and $S_2$, compete for the channel in presence of elastic flows, $E_1$.

Two basic scenarios are considered:

1. A variable number of $S_1$ flows ($n_{s,1}$) and a fixed number of nodes with $S_2$ flows ($n_{s,2} = 2$) (scenario 1).

2. A variable number of $S_1$ flows ($n_{s,1}$), a fixed number of nodes with $S_2$ flows ($n_{s,2} = 2$) and a fixed number of nodes with $E_1$ flows ($n_{e,1} = 2$) (scenario 2).

Figures 4 and 5 report the throughput for the three types of traffic flows in both scenarios. We would like to underline that the model can capture the point where both $S_1$ and $S_2$ flows fail to achieve their bandwidth requirements. In Table 6 the queue utilization of a node is compared with simulation results (scenario 1). Note that the model provides pessimistic values but matches the dynamics of the queue utilization.

In Table 7 we show the conditional collision probability, the expected number of slots of the backoff instance before a transmission attempt, and the channel probabilities observed by a $S_1$ flow.

In the first column of Table 8 we include the queue occupation for $S_1$ nodes in the homogeneous scenario. Note how the introduction of $n_{s,2} = 2$ $S_2$ flows causes an increment of the queue utilization for $S_1$ flows. Therefore, a clear interaction from $S_2$ flows exists and is added to the own interaction between $S_1$ flows. The total interaction, which is non-linear with the number of nodes, makes
the queue utilization of $S_1$ nodes saturate more rapidly. At the same time, for a fixed number of $S_2$ nodes ($n_{s,2} = 2$), we can observe how their queue utilization is also correlated with the queue utilization of $S_1$ flows.

For admission control purposes, it is worth noting that in scenario 1, with $n_{s,1} = 7$ flows, if another $S_1$ flow is accepted, the new accepted flow will perform correctly while the $S_2$ flows will perform poorly. Therefore, if the $S_2$ service degradation is not acceptable, this new $S_1$ should be rejected. Notice also that we cannot evaluate independently the two types of flows because the maximum number of flows for each type must be related to the presence of the other type of flows.

5.4 Model applications

One of the most interesting features of the model is its flexibility to reproduce several situations of interest:

Complex scenarios. As each node can be configured independently, it is easy to model nodes with different functions in the same network (access point, mobile nodes, etc.). This flexibility was difficult to achieve using saturated models due to their reduced parametrization.

Multi-rate capabilities. The extension of the model to multi-rate networks is rather straightforward, as we can assign to each flow/node a different value of $R_{data}$.

Admission control and resource scheduling. As the model catches the relationships between flows, it can be used to evaluate different resource strategies, such as those based on setting the $CW$ value, and test the overall effect of this setting on the network performance.

6 User-level performance in Hot-Spot WLANs

The model presented in previous section is applied to a real scenario: an infrastructured WLAN network (BSS). First, using the ns2 package [26] as simulation tool we investigate the performance of a basic WLAN cell with two types of traffic: a) TCP-like traffic, i.e., persistent connections using the RENO version of the TCP protocol, and b) VoIP CBR sources. Both simulation and analytical results are derived considering the use of the RTS/CTS mechanism. Performance are evaluated in terms of MAC layer throughput which includes the upper layers overhead. To clarify the notation,
we refer to every parameter related with the AP with the subscript $d$ (downlink) and with the subscript $u$ (uplink) we refer to the parameters of the mobile nodes. The number of elastic flows in the downlink (uplink) will be denoted by $n_{e,d}$ ($n_{e,u}$) and the number of VoIP calls by $n_s$.

Currently, some papers address the issue of analytically modeling the TCP throughput performance in WLAN networks. Basically, for the downlink direction, Bruno et al. [12], Miorandi et al. [13] and Lebeugle et al. [14] present several models to compute the WLAN throughput but with similar theoretical basis. In the uplink direction, a model is presented by Leith et al. [28] with also similar applicability. For what concerns TCP flows simultaneously in both directions, Pilosof et al. [29], explain the major observed phenomena but, to the best of our knowledge, the only work which treats it analytically is presented by Bruno et al. [15] under the assumption that the TCP advertisement window is equal to one.

### 6.1 A general view of TCP performance

WLAN cells (or hot-spots) based on the IEEE 802.11 technology are deployed massively in business or public areas. From [30], more than 90% of the total traffic is TCP-based. Nowadays, most of the traffic is due to HTTP transactions; however, it is worth mentioning that the P2P traffic reaches significant levels, higher than those obtained by the email or the FTP services. Another interesting observation is the asymmetry of the traffic flows, where the 85% goes from the fixed network to mobile nodes (downlink) and the remaining 15%, which is a significant value, from mobile nodes to the fixed network (uplink).

One of the most important questions for hot-spot operators is to know the grade of service that an user receives. This grade of service, in case of elastic traffic, can be measured in terms of delay to visualize the data object requested (a web page, a photograph, etc.), the time spent to receive or send an email with an attached file, to transfer some files to another computer using FTP, a P2P transfer, etc. This delay is directly related to the bandwidth used by each flow. Therefore, the knowledge of the system performance can be used to provide service guarantees to users with different traffic profiles. For example, in the presence of users with a minimum bandwidth requirement and a set of users with a pure best-effort policy, admission and rate control strategies should be used to meet the requirements.

In order to compute the performance of the wireless cell, the TCP protocol behavior should
be considered. However, a detailed model of the TCP protocol is a hard and complex task which is out of the scope of this work. We therefore suggest a simple analysis based on the assumption that a node with a TCP flow can be modeled as a saturated queue. This assumption, as we show, allows us to obtain accurate results in terms of steady-state performance.

A first simulation result consisting of only downlink $S_{e,d}^{tcp}$ and uplink $S_{e,u}^{tcp}$ TCP flows is shown in Table 9. The maximum TCP window size has been fixed to $W = 1$ and $W = 42$ (as it is commonly used in the operative TCP versions [29]). Two basic system throughput tendencies can be underlined:

- The increment of the number of downlink TCP flows does not reduce the aggregated throughput due to the fact that TCP reduces the channel contention [12] (the average number of backlogged nodes with feedback traffic is lower than the number of TCP flows).

- As we increase the number of TCP flows in the uplink, the aggregate throughput is increased since the TCP window of mobile nodes reaches its maximum value (for $W > 1$) despite packet losses and the starvation of the downlink ACK flow [28, 29]

However, previous mentioned works also show that it exists unfairness among TCP flows in both uplink and downlink directions. For the sake of simplicity, we assume that all flows share fairly the aggregate throughput, then each flow receives $S_{e,z}^{tcp} / n_{e,z} \text{ bps}$, with $z = \{d, u\}$.

### 6.1.1 Downlink TCP flows

In the downlink, TCP flows compete, through the AP, with their own feedback traffic sent by the mobile nodes.

A single downlink TCP flow, with maximum window size equal to $W = 1$ has a throughput proportional to $L_{tcp}/RTT$, where $RTT$ is the TCP Round Trip Time. Then, the throughput is computed from

$$S_{e,d=1}^{tcp}(W = 1) = \frac{L_{tcp}}{1000e6} + X_d(L_{tcp}) + \frac{L_{ack}}{1000e6} + X_u(L_{ack}) + 2\delta$$

(18)

where $X_d(L_{tcp})$ ($X_u(L_{ack})$) is the service time over the WLAN for a TCP (ACK) packet and $\delta$ is the signal propagation delay. Simulation ($S_{e,d=1}^{tcp}(W = 1) = 0.896 \text{ Mbps}$) and analytical results
\( S_{e,d=1}^{\text{tcp}}(W = 1) = 0.879 \text{ Mbps} \) show, as we expect, a good match.

With \( W = 1 \) and a single TCP flow, there is no competition to access the channel between the AP and the mobile nodes because the two nodes never simultaneously have a packet ready to be transmitted at the MAC queue. From Table 9, as the number of simultaneous TCP flows increases, despite keeping \( W = 1 \), the AP queue tends to have always a packet ready to be transmitted, which justifies the assumption to model the access point as a saturated queue \([12, 13, 14]\), which is obviously confirmed for values of \( W > 1 \).

For each received packet, a mobile node sends the correspondent ACK (we have not considered the delayed ack technique as in \([13]\)). Therefore, the number of ACK packets sent by a mobile node will be \( 1/n_{e,d} \) per packet emitted by the AP.

To analyze this situation, several approximations can be used.

**All sources are saturated (Model \( A_d \))**. In this model, both the AP and the mobile nodes are considered as saturated with data and ACK packets, respectively. This approximation provides pessimistic results because the level of contention suffered by data packets is very high as the AP has to compete with all the mobile nodes.

**A time-scale decomposition (Model \( B_d \)) \([12, 13]\)**. Introduced by \([12]\) and used also by \([13]\), this model computes the distribution of backlogged nodes \( n^b_{e,d} \), i.e., the probability that \( n^b_{e,d} \) of the \( n_{e,d} \) mobile nodes are backlogged. The system throughput is computed averaging the throughput obtained with \( n^b_{e,d} \) saturated nodes for \( n^b_{e,d} = 0 \ldots n_{e,d} \). We suggest a novel variant of this model, where transitions between states are done after any successful transmission in the channel and not only after a successful transmission of the AP as in \([12]\). In Figure 6 we show the DTMC (Discrete-Time Markov Chain) which governs the number of backlogged nodes in function of the number of downlink TCP flows, \( C \). Note that the DTMC changes its state after any successful transmission over the channel, independently on that it has been done by the AP or a mobile node. The probability to move from state \( j-1 \) to state \( j \) depends on the probability that the AP transmits, \( 1/j \), and the probability that the packet was sent to a non-backlogged mobile node, \((C-j+1)/C\). The probability to remain in the same state \( j \) is the probability that the AP transmits a packet, \( 1/(j+1) \), which is sent to a backlogged node with probability \( j/C \). Finally, the probability to move from state \( j \) to state \( j-1 \) is the
probability that a backlogged mobile node transmits, $j/(j+1)$. Note that a single ACK is stored in each mobile node queue.

**Un-correlated ACKs (Model $C_d$)**. Finally, by simply applying the MAC model described in previous section, a good approximation can also be obtained. In this case, we configure the arrival rate of the mobile nodes as $\lambda_{e,u} = \lambda_{e,d}/n_{e,d}$ where $\lambda_{e,d}$ is computed as the maximum arrival rate of the AP under the condition that $\nu_d = 1$ and assuming the TCP packets are being distributed uniformly among destination nodes. Is worth noting that we can model the delayed ACKs technique by simple divide the value of $\lambda_{e,u}$ by the delayed ACK factor $\gamma$, $\lambda_{e,u} = \lambda_{e,d}/(\gamma \cdot n_{e,d})$

In Table 10 the downlink throughput is obtained by simulation and compared with the outcomes of the three models previously described. Note how model $A_d$ clearly overestimates the negative effects of the feedback traffic and thus, the throughput obtained is lower than in simulation. Model $B_d$ provides a very good approximation. Finally, model $C_d$ also shows very accurate results, despite the assumption of Poisson arrivals for the ACK packets, that corresponds to assuming that there is no correlation with the reception of TCP data packets. These results allow us to validate our MAC model in this new scenario.

### 6.1.2 Uplink TCP flows

TCP flows in the uplink compete among themselves and with the feedback traffic from the AP ACKs. Leith et al. [28] show the existing unfairness among competing uplink TCP flows. They also propose an analytical model for the uplink TCP throughput addressing an ACK prioritization at the AP using the EDCF (Enhancement Distributed Coordination Function [4]) to reduce the inherent asymmetry of the WLAN. The asymmetry is due to the fact that mobile nodes gain the $n_{e,u}\rho_u/(n_{e,u}\rho_u + \rho_d)$ transmission opportunities to transmit TCP data packets and the AP only gains $\rho_d/(n_{e,u}\rho_u + \rho_d)$ for the ACK packets. Their model assumes that mobile nodes are saturated and the AP is not. To compute the transmission probability of the AP they use the fact that all TCP data packets are answered by a single ACK packet. Then, by analogy with the model presented in this paper, the transmission probability of the access point is the probability that the access point observes a successful transmission in the channel, i.e., $\tau_d = p_{s,d}$, with $p_{s,d}$ computed as...
in Eq. (17). This first model is called Model $A_u$.

In Table 11 we compare the results obtained by simulation for $W = 42$ and with the results obtained by previous Model $A_u$ and an additional model which assumes that the AP is also saturated (Model $B_u$). Both models provide good accuracy.

### 6.1.3 Simultaneous Downlink and Uplink TCP flows

Finally, when there are multiple TCP flows in both directions, $n_{e,d}$ in the downlink and $n_{e,u}$ in the uplink, the performance of downlink flows is severely affected, as we can see in Table 12. In this configuration, the AP queue is shared by both ACKs and data packets while mobile nodes only send either TCP data packets or ACKs (in the system there are $n_{e,u}$ nodes sending TCP data packets and $n_{e,d}$ sending ACKs). The results confirm those in [29] about the different behavior of the TCP window for the uplink and downlink flows. Pilosof et al. argue that the TCP window for uplink senders reaches the maximum value, even with high ACK losses at the AP buffer, while downlink flows struggle with low window values ($0 - 2$ packets) caused by frequent timeouts due to data packet drops.

For $W = 1$ and a low number of upstream and downstream flows the $RTT$ is relatively independent on whether the TCP data packet is sent by a mobile node or a server in the fixed network, and, thus, the uplink and downlink performance is the same. Beyond $n_{e,d} = n_{e,u} = 6$, differences between the streams become perceptible and the tendency of the downlink throughput $S_{tcp}^{e,d}$ to decrease becomes clearly visible while the $S_{tcp}^{e,u}$ throughput continues growing. However, when $W > 1$, the results show how that the uplink TCP flows achieves much higher throughput than the downlink flows.

Modeling this situation is very complex due to the interaction in the AP queue of the two types of packets: ACKs and data packets. We suggest a simple approximation (model $A_b$) which captures the main tendencies observed in the simulations:

- The downlink queue is always saturated. The average packet length transmitted by the AP is computed from $EL_d = \phi_d L_{tcp} + \phi_u L_{ack}$ where $\phi_d$ and $\phi_u$ are the probability that a packet sent by the AP is a data or an ACK packet.

- These probabilities are computed from: $\phi_d = 1 - \phi_u$ and $\phi_u = n_{e,u} / (n_{e,u} + n_{e,d})$. Note that, if
\( n_{e,d} = 0 \), this model is equivalent to the model used for the uplink (model \( B_u \)) and if \( n_{e,u} = 0 \), the model is equivalent to the model used for the downlink (model \( C_d \)).

- Mobile nodes with uplink data packets are always saturated.
- Nodes with uplink ACKs have \( \lambda_{ack} = \lambda_{e,d}/n_{e,d} \), where \( \lambda_{e,d} = \phi_d/X_d(EL_d) \).

At the access point queue ACKs controlled by the transmission opportunities of mobile nodes compete with data packets of downlink flows. As the number of uplink flows increases, since mobile nodes have more transmission opportunities than the AP, the number of ACKs in the AP increases. Thus, the downlink flows suffer for both contending for buffer space in the AP, and for contending on the access to the channel. Downlink TCP flows tend to starve, as can be noticed from results in Table 12.

### 6.2 Voice over IP

Voice communication using WLAN technology as access network could be a promising alternative to traditional cellular networks (2G, 3G). Currently, roaming problems between WLAN coverage areas have to be solved to provide a continuous service to the user. However, novel proposals to interconnect and manage WLAN cells using common fixed infrastructure operators are yet a promising reality [31]. Moreover, three major technological issues of the IEEE 802.11 MAC protocol itself have to be solved or improved to achieve an efficient use of the transmission resources.

1. High protocol overheads.
2. Unfairness between uplink and downlink streams.
3. Fast VoIP degradation in presence of TCP flows.

A criterium to determine the maximum number of VoIP calls that can be transported by a network (also called VoIP capacity) given the desired voice quality in terms of bandwidth, delay, losses, can be found in [32]. For a good quality, the average delay have to be less than 150 ms with losses less than 3%. A medium quality is achieved with delays between 150 and 400 ms and packet losses less than 7%. Finally, a poor voice quality corresponds to delays higher than 400 ms and losses higher than 7%. Considering that the WLAN is only one hop of the whole path between the
two end points, for a conservative good design, quality target should be set to at least one-third of the maximum recommended values.

In Table 13 we summarize the basic characteristics of the most frequently used voice codecs for VoIP. The average throughput is plotted in Figure 7 for the G.711 and G.729 voice codecs. The AP is the bottleneck of the system, which limits the VoIP capacity.

6.2.1 Protocol Overheads

Taking into account the parameters defined by the IEEE 802.11b standard [1], summarized in Table 1, we can compute the maximum number of voice calls without contention, i.e., the channel is ideally shared among voice calls, and with contention, see results in Table 14. From the results, we can conclude that the contention to access the channel reduces the maximum number of calls but it is not the main limiting factor. It is clear that the main problem is the large overhead introduced by the higher layer protocols. A technique to solve this situation is header compression such as ROHC (RObust Header Compression, RFC 3243).

6.2.2 Unfairness

As the access point carries the same data as the whole set of the mobile nodes, it has to attempt to transmit \( n \) times more than each mobile node. Therefore, it is desirable that the transmission attempts of the AP are \( n \) times greater than the transmission attempts of a single mobile node, or equivalently \( \tau_d = n \tau_u \), to achieve a fair access to the channel (each node access to the channel proportionally to the traffic volume it has to send).

As a measure of the system fairness, we compare

\[
\begin{align*}
  w_d &= \frac{\tau_d}{n_u \tau_u + \tau_d} \\
  w_u &= \frac{n_u \tau_u}{n_u \tau_u + \tau_d}
\end{align*}
\]  

Considering the G.729 voice codec, the first two columns of Table 15 show \( w_d \) and \( w_u \) versus the number of mobile nodes. As the number of voice calls (or mobile nodes) increases, the AP unfairness grows, resulting in a fast saturation of the AP queue that limits the system capacity. To solve this problem, a simple solution consists in updating the \( CW_{\text{min}} \) value of mobile nodes each time a new call arrives at the system. By computing the value of \( CW_{\text{min}}^* \) for each mobile node,
assuming $m = 0$, we obtain

$$CW_{min,u}^* = \frac{n_s \rho_u (CW_{min,d} + 1)}{\rho_d} - 1$$

(20)

The fairness obtained by applying this solution is showed in the two last columns of table 15. Note that the AP gets equal or more transmission opportunities than the uplink mobile nodes.

This solution can also be combined with the one presented in [17], where the TXOP mechanism of EDCF [4] is used to provide fairness. However, in both cases a limited gain is obtained in terms of capacity increment. Finally, another solution, similar to the use of TXOP, is presented in [18]. In this case, several voice packets are encapsulated in only one multicast packet which is sent to all mobile nodes, where each one gets its own data.

### 6.2.3 Interaction with TCP flows

As the AP queue is shared by all downlink streams, the VoIP packets have to compete for the buffer space with all the other flows, that can be streaming (UDP) or elastic (TCP) flows (both data and ACKs destined to a mobile node). In previous section, we have concluded that the TCP traffic tends to saturate the MAC queue and cause high losses if it is shared with VoIP packets. However, uplink flows reduce the transmission opportunities gained by the AP, especially if they are uplink TCP flows which also tend to saturate the mobile nodes queue.

The negative influence of the TCP traffic is clearly showed in Table 16. Note the fast degradation of the VoIP throughput with TCP downlink flows and the inoperability of any VoIP call with just a single TCP uplink flow. It is also interesting to observe that, when the AP queue is saturated with VoIP traffic, the interaction with TCP traffic is reduced due to the starvation of TCP flows.

Therefore, the presence of TCP traffic in both the downlink (buffer losses) and the uplink (AP starvation) leads to low performance of VoIP calls. These problems have to be solved in order to deploy a successful VoIP service over WLAN.

In the downlink, a simple classification/prioritization scheme can be used (for example the dual queue proposed in [33]) where the TCP and UDP packets occupy separated buffers. However, the main problem is with the TCP uplink flows because each node acts independently from the others. The only possible solution is setting different MAC parameters (such as $CW_{min}$, $R$, $m$, TXOP) to
each mobile node in order to reduce the interaction of these TCP flows on the VoIP calls.

7 Call Admission Control

With the goal to solve the performance problems previously exposed about the interaction between TCP and UDP traffic, we propose an admission control scheme which is capable to differentiate between streaming and elastic traffic and, at the same time, between uplink and downlink flows, providing an acceptable grade of service for streaming flows (VoIP).

7.1 CAC Architecture

The call admission control entity is located at the AP. When an application wants to use the cell resources it sends a request packet (for example it could be similar to the Add Traffic Specification, ADDTS, packet [4]) to the AP with the traffic profile required by the flow. Using the information provided by the application, the call admission control decides if the new state of the network is feasible. If so, it sends a positive response to the request. Otherwise, it sends a negative response and the new flow is rejected, preserving the grade of service of the active flows already in the system.

To differentiate TCP and UDP downstream flows, the AP uses a dual queue [33], where UDP packets are isolated from TCP packets and with a service prioritization for the UDP queue. Moreover, the upstream flows are differentiated by setting different MAC layer parameters ($CW_{min}$) for each flow.

7.1.1 A dual queue scheme at the Access Point

Similar to the case in [33], we propose a dual queue strategy to differentiate between downstream TCP and UDP packets. We refer with $\rho_{s,d}$ to the UDP queue utilization and with $\rho_{e,d}$ to the TCP queue utilization, respectively. TCP packets are served only when the UDP queue is empty, then the probability to transmit a downstream TCP packet is $1 - \rho_{s,d}$. Note that, since the upstream feedback traffic is proportional to the downstream TCP traffic, we can assume a minimal impact of uplink TCP ACKs over the UDP packets.

Without considering the presence of TCP uplink flows at the moment, the downlink TCP
throughput can be measured as a function of the UDP queue utilization as \( ES_{e,d} = (1 - \rho_{s,d})S_{e,d}^{tcp} \), by using the model presented in previous section (model \( C_d \)). Clearly, the elastic flows can suffer starvation when \( \rho_{s,d} \approx 1 \). However, this can be solved by setting a maximum \( \rho_{s,d}^{th} \) value, e.g., \( \rho_{s,d}^{th} = 0.8 \), which assures that at least the \( 1 - \rho_{s,d}^{th} \) of the transmissions are for TCP downstream flows.

### 7.1.2 Uplink differentiation via different values of \( CW_{min} \)

In order to differentiate the VoIP flows with respect to the TCP uplink flows (which cause a major performance degradation), we propose to use different \( CW_{min} \) values that prioritize streaming flows over elastic flows. Let \( \Psi_{CW} \) be the set of all possible \( CW_{min} \) values that can be used by uplink elastic flows, with \( \Psi_{CW} = \{32, 64, 128, 256, 1024\} \). When the CAC receives a new request of a TCP uplink flow, it computes the suitable \( CW_{min} \) value for the new and all the already active uplink elastic flows and broadcasts the new \( CW_{min} \) values. If there are no VoIP flows in the system we assume that all nodes and the AP use the standard value of \( CW_{min} \).

### 7.2 Proposed algorithm

We propose an algorithm based on the estimates provided by the model presented in previous section to predict the network behavior when a new flow request is received. Notice that the estimates can be either provided by another model or based on measurements. However, we believe that the proposed model is a good trade-off between simplicity and accuracy. Each request is configured to provide at least the information: 1) flow type (F), that can be elastic or streaming, 2) requested bandwidth (B), and 3) average frame length (L). Using these pieces of information, the suggested algorithm operates as follows,

1. A new request is received by the admission control with parameters \((F, B, L)\).

2. Using the current system information plus the new flow request, the new system state is estimated.

   - If the new request is for a downlink elastic flow, it is accepted if the number of downlink elastic flows is lower than the threshold \( N_{e,d}^{th} \). The elastic downlink throughput is auto-regulated by the dual-queue mechanism and the TCP dynamics.
• If the new request is for an uplink elastic flow, it is accepted if the number of uplink elastic flows is lower than the threshold \( N_{e,u}^{th} \) and if the new state is feasible. If it is not feasible, it tests from the set of \( CW_{min} \) values if using another \( CW_{min} \) value for the elastic flows, the new state is possible.

• If the new request is for a downlink streaming flow, the CAC evaluates the queue utilization of the downlink UDP queue. If \( \rho_{s,d}^{\ast} < \rho_{s,d}^{th} \), the new flow is accepted. The \( \rho_{s,d}^{\ast} \) parameter is estimated considering also the presence of the new flow. After accepting the new flow, the current \( \rho_{s,d} \) is also updated with the information of the new flow.

• If the new request is for an uplink streaming flow, the CAC evaluates the system state: if the new state is feasible, it accepts the flow. If not, it tests if another combination of \( CW_{min} \) for elastic flows can make the new state feasible.

• If the new request is for a bidirectional streaming flow, the system evaluates if both uplink and downlink flows can be accepted by using previous explanations.

3. If no state is feasible, reject the new flow.

7.3 Performance Evaluation

7.3.1 Model of a cell

Four types of flows are considered: downlink / uplink streaming flows and downlink / uplink elastic flows. Under the assumption of exponential distributions of flow arrivals and departures, the system can be described by a Continuous Time Markov Chain (CTMC). If we also assume that all VoIP calls comprise one uplink and one downlink flow and use the same codec, then the state of the CTMC is given by the vector \( (n_{e,d}, n_{e,u}, n_s) \) where \( n_{e,d} \), \( n_{e,u} \) and \( n_s \) denote the number of elastic flows (downlink and uplink) and VoIP calls that are active in the cell. To solve this CTMC we suggest to break the three-dimensional CTMC in two bi-dimensional CTMCs. First CTMC \( (CTMC_A) \) comprises the situation where the VoIP calls compete with uplink TCP flows and second CTMC \( (CTMC_B) \) the situation where downlink TCP flows compete with uplink TCP flows. The partial results of both CTMC can be averaged using the approximation that with probability \( \rho_{s,d} \) the system works in the situation described by \( CTMC_A \) and with probability \( 1 - \rho_{s,d} \) the system behavior can be modeled by \( CTMC_B \).
While the number of elastic flows can grow to infinity, the maximum number of streaming flows is limited by the bandwidth requirements of the voice calls to $N_{\text{voip}}^{th}$. In order to solve these infinite bi-dimensional CTMC, we need to truncate the state space. Without loss of generality, we introduce a realistic minimum bandwidth $B_{e,\text{min}}$ required for an elastic flow which gives a maximum number of $N_{e,u}^{th}$ ($N_{e,d}^{th}$) uplink (downlink) elastic flows. The CTMC state space is described by

$$CTMC_A : S_A = \{(n_{e,u}, n_s) | S_{e,u}^{\text{tcp}}(n_{e,u}, n_s)/n_{e,u} \geq B_{e,\text{min}}, S_{e,u}^{\text{voip}}(n_{e,u}, n_s) > 0.97 \cdot n_s B_s\}$$

$$CTMC_B : S_B = \{(n_{e,u}, n_{e,d}) | S_{e,u}^{\text{tcp}}(n_{e,u}, n_{e,d})/n_{e,u} \geq B_{e,\text{min}}, S_{e,d}^{\text{tcp}}(n_{e,u}, n_{e,d})/n_{e,d} \geq B_{e,\text{min}}\}$$

(21)

For both voice and elastic flows the user population is considered to be infinite with steady state arrival rates $\lambda_{e,u}$ and $\lambda_{e,d}$ for elastic flows and $\lambda_s$ for VoIP calls. The elastic flow duration is function of the bandwidth observed by the elastic flows and the flow length (amount of data to transmit) $F_{L,e}$, with departure rate equal to $\mu_{e,x} = S_{e,x}^{\text{tcp}}(\cdot)/(n_{e,x} F_{L,e})$, and $\mu_s$ for streaming flows, which have a fixed average duration.

7.3.2 Parameters

The goal of this section is to evaluate the effect of elastic uplink flows over the capacity of VoIP calls. The parameters considered to test our CAC algorithm are the following:

1. Voice Calls

   - Voice codec: G.729 with $L = 20$ Bytes and rate equal to 8 Kbps.
   - We assume that the voice call requests follow a Poisson process with rate $\lambda_s = 0.0083 \text{ calls/s}$ (one call every two minutes), and that the duration of a call is exponentially distributed with mean $1/\mu_s = 240$ s (four minutes).

2. Elastic Flows

   - The arrival process of elastic flows is also assumed to be Poisson with rates $\lambda_{e,d} = 1$ and $\lambda_{e,u}$ flows/second. Since we wish to evaluate the impact of uplink elastic flows over VoIP calls, the parameter $\lambda_{e,u}$ will vary.
Two elastic flow lengths are considered: $FL_e = 1 \text{ Mbits}$ and $FL_e = 2 \text{ Mbits}$, with exponential distribution (equal values for the downlink and the uplink).

For the sake of simplicity, $B_{e,min}$ is computed to allow a maximum number $N_{e,u}^{th} = N_{e,d}^{th} = 10$ of active elastic flows in the system.

### 7.3.3 Considered metrics

In order to evaluate the proposed mechanism, we compute the following performance metrics,

1. **Voice Calls**
   
   - Blocking Probability: $BP_s$

2. **Elastic Flows**
   
   - Uplink (Downlink) Blocking Probability: $BP_{e,u}$ ($BP_{e,d}$)
   - Uplink (Downlink) Elastic Throughput: $ES_{e,u}$ ($ES_{e,d}$)

The analysis of the previous metrics is straightforward from the CTMC which defines the system state. The averaged downlink throughput is measured as

$$ES_{e,d} = (1 - \rho_{s,d})E[S_{ tcp}^{el}(n_{e,u}, n_{e,d})]$$

where $E[S_{ tcp}^{el}(n_{e,u}, n_{e,d})]$ is the averaged downlink throughput computed from $CTMC_B$. The averaged uplink elastic throughput can be approximated by

$$ES_{e,u} = \rho_{s,d}E[S_{ tcp}^{el}(n_{e,u}, n_{s})] + (1 - \rho_{s,d})E[S_{ tcp}^{el}(n_{e,u}, n_{e,d})]$$

where $E[S_{ tcp}^{el}(n_{e,u}, n_{s})]$ and $E[S_{ tcp}^{el}(n_{e,u}, n_{e,d})]$ are computed respectively from $CTMC_A$ and $CTMC_B$. The values $S_{ tcp}^{el}(n_{e,u}, n_{e,d})$, $S_{ tcp}^{el}(n_{e,u}, n_{s})$ and $S_{ tcp}^{el}(n_{e,u}, n_{e,d})$ are computed using the models of previous section. Similar approximation is applied to compute the blocking probability, assuming that all downlink TCP flows are blocked if some VoIP conversations are active.
7.4 Numerical Results

In this section we present some numerical results to investigate the performance of the proposed CAC scheme. Results are shown as the comparison of the performance achieved by the WLAN using a non adaptive CAC called \textit{simple CAC}, according to which all flows always use a fixed value of $CW_{\text{min}}$ equal to 32, and using the proposed CAC scheme, called \textit{adaptive CAC}.

Figure 8 shows the voice call blocking probability $BP_s$ for two different elastic flow lengths, with the \textit{simple CAC} and the \textit{adaptive CAC}. For both CAC schemes, as we expect, the blocking probability increases as the traffic intensity of uplink elastic flows grows. However, under the \textit{adaptive CAC} the blocking probability exhibits substantially lower values, due to the growth of the number of feasible states deriving from the reduction of the transmission rate of elastic flows (due the increment of the value of their $CW_{\text{min}}$ parameter) which decreases the possibility that the arrival of a new voice call is blocked. Observe that for values of $\lambda_{e,u}$ greater than $\lambda_{e,u} = 1.5$ ($FL_e = 2 \text{ Mbits}$) ($\lambda_{e,u} = 3.25$ ($FL_e = 1 \text{ Mbits}$)), using the simple CAC scheme all voice calls are blocked, while using the \textit{adaptive CAC} scheme, this probability remains rather constant and about 0.2.

Previous results show the goodness of the \textit{adaptive CAC} to reduce the blocking probability of voice calls. However, it has negative effects on the elastic flows, increasing the blocking probability of them and reducing their throughput. Figure 9 reports the blocking probability for uplink elastic flows. Two regions can be observed: a) the blocking probability of elastic flows is mainly caused by voice calls ($\lambda_{e,u} \leq 0.5$, $FL_e = 2 \text{ Mbits}$) or ($\lambda_{e,u} \leq 1.5$, $FL_e = 1 \text{ Mbits}$) and b) the blocking probability of elastic flows is caused by their own traffic. In the first region, for low traffic intensity of elastic flows, the blocking probability using the \textit{adaptive CAC} is reduced as the number of uplink elastic flows that can coexist with the VoIP calls grows. However, in the second region, as the average number of active elastic flows is higher, the blocking probability of voice calls is near 1 (see Figure 8), so the presence of active voice calls is negligible, the blocking probability of elastic flows is due mainly to its own traffic. Using the \textit{adaptive CAC}, which preempts the elastic flows (reducing their transmission rate), the blocking probability of elastic flows increases as the output rates for uplink elastic flows are reduced, which results in values higher than the \textit{simple CAC}. As a conclusion, the \textit{adaptive CAC} is significantly better for both VoIP calls and elastic flows than the
simple CAC when the uplink elastic flow intensity is not enough to saturate the system. Otherwise, for high elastic flow intensity, the scheme maintains low the blocking probability of VoIP calls while the performance of elastic flows is not severely reduced.

The higher number of accepted VoIP calls using the adaptive CAC results in an increment of the VoIP throughput and then, a higher utilization $(\rho_{s,d})$ of the downlink streaming queue at the AP, decreasing the transmission opportunities of the TCP downlink traffic. This situation entails an increase of the blocking probability of downlink TCP flows (Figure 10).

Finally, observe the uplink / downlink elastic throughput using the simple and the adaptive CAC schemes (Figures 11 and 12) which show coherent results with previous explanations.

From the presented numerical results we can conclude that if a CAC is implemented for WLAN networks to provide QoS to multimedia traffic flows, it has to modify the MAC parameters of mobile nodes in order to optimize the overall WLAN performance.

8 Conclusions

In this chapter we have discussed some issues related to the provision of integrated services through WiFi access networks. The popularity of WiFi access and the common use of a variety of different services is making this topic the more and more crucial for operators in the field.

In particular, the performance of the IEEE 802.11 MAC protocol (DCF) was investigated by means of a novel user-centric model that was validated against simulation results. The model was used to analyze the case of heterogeneous traffic scenarios, in which elastic and streaming traffic, representing respectively TCP-based applications and VoIP service, share the common radio resources. The interaction between the two classes of traffic was highlighted.

The analysis suggested that, in order to provide good quality of service to traffic flows whose requirements are so different to each other, call admission control is needed. A novel scheme was then proposed based on both the use of the network state and a smart setting of the MAC layer parameters.
9 Acknowledgements

This work was partially supported by Catalanian Government under the *i2CAT* (Advanced Internet in Catalonia) project, Spanish Government under project *TIC2003-09279-C02-01*, and by the European Commission under *NEWCOM* network of excellence.
References


<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{data}$</td>
<td>2 Mbps</td>
<td>$R_{basic}$</td>
<td>1 Mbps</td>
</tr>
<tr>
<td>DIFS</td>
<td>50 µs</td>
<td>$CW_{min}$</td>
<td>32</td>
</tr>
<tr>
<td>SIFS</td>
<td>10 µs</td>
<td>$CW_{max}$</td>
<td>1024</td>
</tr>
<tr>
<td>SLOT ($\sigma$)</td>
<td>20 µs</td>
<td>$m$</td>
<td>5</td>
</tr>
<tr>
<td>EIFS</td>
<td>364 µs</td>
<td>ACK</td>
<td>112 bits @ $R_{basic}$</td>
</tr>
<tr>
<td>RTS</td>
<td>160 bits @ $R_{basic}$</td>
<td>CTS</td>
<td>112 bits @ $R_{basic}$</td>
</tr>
<tr>
<td>MAC header</td>
<td>240 bits @ $R_{data}$</td>
<td>MAC FCS</td>
<td>32 bits @ $R_{data}$</td>
</tr>
<tr>
<td>PLCP preamble</td>
<td>144 bits @ $R_{basic}$</td>
<td>PLCP header</td>
<td>48 bits @ $R_{basic}$</td>
</tr>
<tr>
<td>Retry Limit (R)</td>
<td>7</td>
<td>$Q$ (Queue length)</td>
<td>50 packets</td>
</tr>
</tbody>
</table>

Table 1: System parameters of the IEEE 802.11b specification [1]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTP header</td>
<td>12 Bytes</td>
</tr>
<tr>
<td>TCP header</td>
<td>20 Bytes</td>
</tr>
<tr>
<td>UDP header</td>
<td>8 Bytes</td>
</tr>
<tr>
<td>IP header</td>
<td>20 Bytes</td>
</tr>
</tbody>
</table>

Table 2: Protocol overheads from upper layers

<table>
<thead>
<tr>
<th>Traffic Flow</th>
<th>Bandwidth</th>
<th>Frame Length</th>
<th>Retry Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>elastic (E1)</td>
<td>max.available</td>
<td>1500 Bytes</td>
<td>7</td>
</tr>
<tr>
<td>$streaming$ type 1 (S1)</td>
<td>100 Kbps</td>
<td>400 Bytes</td>
<td>7</td>
</tr>
<tr>
<td>$streaming$ type 2 (S2)</td>
<td>250 Kbps</td>
<td>700 Bytes</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 3: Model validation. Traffic profiles.

<table>
<thead>
<tr>
<th>Model</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{s,1}$</td>
<td>$p_i$</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>1</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0.0052</td>
</tr>
<tr>
<td>4</td>
<td>0.0185</td>
</tr>
<tr>
<td>6</td>
<td>0.0372</td>
</tr>
<tr>
<td>8</td>
<td>0.0663</td>
</tr>
<tr>
<td>10</td>
<td>0.1227</td>
</tr>
<tr>
<td>12</td>
<td>0.3188</td>
</tr>
</tbody>
</table>

Table 4: S1 - Homogeneous case. Model validation for several performance parameters

<table>
<thead>
<tr>
<th>Model</th>
<th>Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_{s,2}$</td>
<td>$p_i$</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>1</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0.0120</td>
</tr>
<tr>
<td>4</td>
<td>0.0560</td>
</tr>
<tr>
<td>6</td>
<td>0.2066</td>
</tr>
<tr>
<td>8</td>
<td>0.2534</td>
</tr>
<tr>
<td>10</td>
<td>0.2897</td>
</tr>
<tr>
<td>12</td>
<td>0.3191</td>
</tr>
</tbody>
</table>

Table 5: S2 - Homogeneous case. Model validation for several performance parameters
### Table 6: $S_1, S_2$ - Queue utilization, scenario 1

<table>
<thead>
<tr>
<th>$n_{s,1}$</th>
<th>$\rho_i$ (Model)</th>
<th>$\rho_i$ (Simulation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1308</td>
<td>0.1130 0.2005</td>
</tr>
<tr>
<td>2</td>
<td>0.1471</td>
<td>0.1210 0.2118</td>
</tr>
<tr>
<td>4</td>
<td>0.1983</td>
<td>0.1559 0.2563</td>
</tr>
<tr>
<td>6</td>
<td>0.3255</td>
<td>0.2441 0.3634</td>
</tr>
<tr>
<td>8</td>
<td>0.9304</td>
<td>0.8743 0.9954</td>
</tr>
<tr>
<td>10</td>
<td>0.9999</td>
<td>0.9892 0.9996</td>
</tr>
</tbody>
</table>

### Table 7: Conditional collision probability, expected number of slots of each backoff instance and channel probabilities ($S_1$), scenario 2

<table>
<thead>
<tr>
<th>$n_{s,1}$</th>
<th>$p_i$</th>
<th>$EB_i$</th>
<th>$p_{e,i}$</th>
<th>$p_{s,i}$</th>
<th>$p_{c,i}$</th>
<th>$p_i$</th>
<th>$EB_i$</th>
<th>$p_{e,i}$</th>
<th>$p_{s,i}$</th>
<th>$p_{c,i}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1785</td>
<td>19.91</td>
<td>0.8214</td>
<td>0.1656</td>
<td>0.0129</td>
<td>0.1789</td>
<td>19.84</td>
<td>0.8149</td>
<td>0.1643</td>
<td>0.0130</td>
</tr>
<tr>
<td>2</td>
<td>0.2039</td>
<td>20.95</td>
<td>0.7960</td>
<td>0.1858</td>
<td>0.0181</td>
<td>0.2037</td>
<td>20.96</td>
<td>0.7892</td>
<td>0.1839</td>
<td>0.0179</td>
</tr>
<tr>
<td>4</td>
<td>0.2525</td>
<td>23.39</td>
<td>0.7474</td>
<td>0.2221</td>
<td>0.0303</td>
<td>0.2477</td>
<td>23.91</td>
<td>0.7439</td>
<td>0.2163</td>
<td>0.0292</td>
</tr>
<tr>
<td>6</td>
<td>0.2890</td>
<td>25.75</td>
<td>0.7109</td>
<td>0.2472</td>
<td>0.0418</td>
<td>0.2812</td>
<td>26.55</td>
<td>0.7085</td>
<td>0.2409</td>
<td>0.0398</td>
</tr>
<tr>
<td>8</td>
<td>0.3185</td>
<td>28.07</td>
<td>0.6814</td>
<td>0.2659</td>
<td>0.0528</td>
<td>0.3088</td>
<td>28.91</td>
<td>0.6833</td>
<td>0.2570</td>
<td>0.0491</td>
</tr>
<tr>
<td>10</td>
<td>0.3431</td>
<td>30.33</td>
<td>0.6568</td>
<td>0.2805</td>
<td>0.0625</td>
<td>0.3305</td>
<td>31.37</td>
<td>0.6609</td>
<td>0.2710</td>
<td>0.0576</td>
</tr>
</tbody>
</table>

### Table 8: $S_1, S_2$ - Mutual interactions among flows, scenario 1

<table>
<thead>
<tr>
<th>$n_{s,1}$</th>
<th>$\rho_i$ (Homogeneous)</th>
<th>$p_i$ (Homogeneous)</th>
<th>$\rho_i$ (Heterogeneous)</th>
<th>$p_i$ (Heterogeneous)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0813</td>
<td>0.0000</td>
<td>0.1308</td>
<td>0.2190</td>
</tr>
<tr>
<td>2</td>
<td>0.0877</td>
<td>0.0052</td>
<td>0.1471</td>
<td>0.2406</td>
</tr>
<tr>
<td>4</td>
<td>0.1044</td>
<td>0.0185</td>
<td>0.1983</td>
<td>0.3082</td>
</tr>
<tr>
<td>6</td>
<td>0.1295</td>
<td>0.0372</td>
<td>0.3255</td>
<td>0.4732</td>
</tr>
<tr>
<td>8</td>
<td>0.1731</td>
<td>0.0663</td>
<td>0.9304</td>
<td>1.0000</td>
</tr>
<tr>
<td>10</td>
<td>0.2760</td>
<td>0.1227</td>
<td>0.9999</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Table 6: $S_1, S_2$ - Queue utilization, scenario 1

Table 7: Conditional collision probability, expected number of slots of each backoff instance and channel probabilities ($S_1$), scenario 2

Table 8: $S_1, S_2$ - Mutual interactions among flows, scenario 1
<table>
<thead>
<tr>
<th>Flows</th>
<th>( S_{e,d}^{tcp} (W = 1) )</th>
<th>( S_{e,u}^{tcp} (W = 1) )</th>
<th>( S_{e,d}^{tcp} (W = 42) )</th>
<th>( S_{e,u}^{tcp} (W = 42) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.896</td>
<td>0.896</td>
<td>1.285</td>
<td>1.286</td>
</tr>
<tr>
<td>2</td>
<td>1.189</td>
<td>1.265</td>
<td>1.285</td>
<td>1.372</td>
</tr>
<tr>
<td>4</td>
<td>1.282</td>
<td>1.275</td>
<td>1.284</td>
<td>1.455</td>
</tr>
<tr>
<td>6</td>
<td>1.267</td>
<td>1.273</td>
<td>1.284</td>
<td>1.454</td>
</tr>
<tr>
<td>8</td>
<td>1.277</td>
<td>1.267</td>
<td>1.280</td>
<td>1.453</td>
</tr>
<tr>
<td>10</td>
<td>1.265</td>
<td>1.270</td>
<td>1.269</td>
<td>1.455</td>
</tr>
</tbody>
</table>

Table 9: \( S_{e,d}^{tcp} \) and \( S_{e,u}^{tcp} \) - Aggregate throughput for persistent TCP flows (Mbps) - \( L_{tcp} = 1500 \) bytes (including the TCP header)

<table>
<thead>
<tr>
<th>Flows</th>
<th>Simulation</th>
<th>Model ( A_d )</th>
<th>Model ( B_d )</th>
<th>Model ( C_d )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( W = 42 )</td>
<td>( W &gt;&gt; 1 )</td>
<td>( W &gt;&gt; 1 )</td>
<td>( W &gt;&gt; 1 )</td>
</tr>
<tr>
<td>1</td>
<td>1.285</td>
<td>1.271</td>
<td>1.357</td>
<td>1.272</td>
</tr>
<tr>
<td>2</td>
<td>1.285</td>
<td>1.082</td>
<td>1.302</td>
<td>1.257</td>
</tr>
<tr>
<td>4</td>
<td>1.284</td>
<td>0.828</td>
<td>1.260</td>
<td>1.250</td>
</tr>
<tr>
<td>6</td>
<td>1.284</td>
<td>0.673</td>
<td>1.243</td>
<td>1.248</td>
</tr>
<tr>
<td>8</td>
<td>1.280</td>
<td>0.556</td>
<td>1.234</td>
<td>1.247</td>
</tr>
<tr>
<td>10</td>
<td>1.269</td>
<td>0.476</td>
<td>1.228</td>
<td>1.246</td>
</tr>
</tbody>
</table>

Table 10: \( S_{e,d}^{tcp} \) - Comparison of TCP downlink models (Mbps) - \( L_{tcp} = 1500 \) bytes (including the TCP header)

<table>
<thead>
<tr>
<th>Flows</th>
<th>Simulation</th>
<th>Model ( A_u )</th>
<th>Model ( B_u )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( W = 42 )</td>
<td>( W &gt;&gt; 1 )</td>
<td>( W &gt;&gt; 1 )</td>
</tr>
<tr>
<td>1</td>
<td>1.286</td>
<td>1.271</td>
<td>1.271</td>
</tr>
<tr>
<td>2</td>
<td>1.372</td>
<td>1.293</td>
<td>1.418</td>
</tr>
<tr>
<td>4</td>
<td>1.455</td>
<td>1.298</td>
<td>1.500</td>
</tr>
<tr>
<td>6</td>
<td>1.454</td>
<td>1.297</td>
<td>1.526</td>
</tr>
<tr>
<td>8</td>
<td>1.453</td>
<td>1.295</td>
<td>1.538</td>
</tr>
<tr>
<td>10</td>
<td>1.455</td>
<td>1.293</td>
<td>1.544</td>
</tr>
</tbody>
</table>

Table 11: \( S_{e,u}^{tcp} \) - Comparison of TCP uplink performance models (Mbps) - \( L_{tcp} = 1500 \) bytes (including the TCP header)

<table>
<thead>
<tr>
<th>Flows (in each direction)</th>
<th>Downlink</th>
<th>Uplink</th>
<th>Downlink</th>
<th>Uplink</th>
<th>Downlink</th>
<th>Uplink</th>
<th>Downlink</th>
<th>Uplink</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( W = 1 )</td>
<td>( W = 42 )</td>
<td>( W &gt;&gt; 1 )</td>
<td>( W &gt;&gt; 1 )</td>
<td>( W &gt;&gt; 1 )</td>
<td>( W &gt;&gt; 1 )</td>
<td>( W &gt;&gt; 1 )</td>
<td>( W &gt;&gt; 1 )</td>
</tr>
<tr>
<td>1</td>
<td>0.582</td>
<td>0.660</td>
<td>0.200</td>
<td>1.084</td>
<td>0.234</td>
<td>0.907</td>
<td>0.149</td>
<td>1.158</td>
</tr>
<tr>
<td>2</td>
<td>0.631</td>
<td>0.625</td>
<td>0.003</td>
<td>1.364</td>
<td>0.149</td>
<td>1.158</td>
<td>0.149</td>
<td>1.158</td>
</tr>
<tr>
<td>4</td>
<td>0.633</td>
<td>0.630</td>
<td>0.000</td>
<td>1.424</td>
<td>0.086</td>
<td>1.340</td>
<td>0.086</td>
<td>1.340</td>
</tr>
<tr>
<td>6</td>
<td>0.629</td>
<td>0.630</td>
<td>0.000</td>
<td>1.425</td>
<td>0.060</td>
<td>1.412</td>
<td>0.060</td>
<td>1.412</td>
</tr>
<tr>
<td>8</td>
<td>0.626</td>
<td>0.634</td>
<td>0.000</td>
<td>1.452</td>
<td>0.046</td>
<td>1.449</td>
<td>0.046</td>
<td>1.449</td>
</tr>
<tr>
<td>10</td>
<td>0.606</td>
<td>0.651</td>
<td>0.000</td>
<td>1.423</td>
<td>0.037</td>
<td>1.471</td>
<td>0.037</td>
<td>1.471</td>
</tr>
</tbody>
</table>

Table 12: \( S_{e,d}^{tcp}, S_{e,u}^{tcp} \) - Comparison of simultaneous TCP downlink/uplink flows (Mbps) - \( L_{tcp} = 1500 \) bytes (including the TCP header)
Codec | G.711 | G.723.1 | G.726-32 | G.729
-----|------|-------|---------|-------
Bit Rate (Kbps) | 64  | 5.3/6.3 | 32     | 8     |
Framing Interval (ms) | 20  | 30     | 20     | 2x10  |
Payload (Bytes)   | 160 | 20/24  | 80     | 10    |
Packets/sec       | 50  | 33     | 50     | 50    |

Table 13: Typical values of most used codecs in VoIP

Max number of calls: $C_{voip}$ (no contention) | 4 | 9/9 | 5 | 6 |
Efficiency (no contention: $\eta = C_{voip} \cdot B_{voice}/R_{DATA}$) | 12.8% | 2.37%/2.85% | 8% | 2.4% |
Max number of calls (contention) | 4 | 7/7 | 4 | 5 |

Table 14: VoIP efficiency over WLAN

<table>
<thead>
<tr>
<th>Voice calls (G.729)</th>
<th>AP ($w_d$)</th>
<th>Mobile Nodes ($w_u$)</th>
<th>AP ($w_d$)</th>
<th>Mobile Nodes ($w_u$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CW_{min}</td>
<td>CW_{min}</td>
<td>CW_{min}</td>
<td>CW_{min}</td>
</tr>
<tr>
<td>1</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
<td>0.500</td>
</tr>
<tr>
<td>2</td>
<td>0.482</td>
<td>0.517</td>
<td>0.501</td>
<td>0.498</td>
</tr>
<tr>
<td>3</td>
<td>0.466</td>
<td>0.533</td>
<td>0.504</td>
<td>0.495</td>
</tr>
<tr>
<td>4</td>
<td>0.450</td>
<td>0.549</td>
<td>0.506</td>
<td>0.493</td>
</tr>
<tr>
<td>5</td>
<td>0.428</td>
<td>0.571</td>
<td>0.510</td>
<td>0.489</td>
</tr>
<tr>
<td>6</td>
<td>0.352</td>
<td>0.647</td>
<td>0.514</td>
<td>0.485</td>
</tr>
<tr>
<td>7</td>
<td>0.217</td>
<td>0.782</td>
<td>0.515</td>
<td>0.484</td>
</tr>
<tr>
<td>8</td>
<td>0.210</td>
<td>0.789</td>
<td>0.516</td>
<td>0.483</td>
</tr>
</tbody>
</table>

Table 15: Bandwidth share between uplink and downlink VoIP flows

<table>
<thead>
<tr>
<th>VoIP calls (G.729)</th>
<th>No TCP flows</th>
<th>TCP Downlink</th>
<th>TCP Uplink</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>1</td>
<td>0.024</td>
<td>0.023</td>
<td>0.023</td>
</tr>
<tr>
<td>2</td>
<td>0.048</td>
<td>0.047</td>
<td>0.045</td>
</tr>
<tr>
<td>3</td>
<td>0.072</td>
<td>0.071</td>
<td>0.066</td>
</tr>
<tr>
<td>4</td>
<td>0.096</td>
<td>0.094</td>
<td>0.088</td>
</tr>
<tr>
<td>5</td>
<td>0.120</td>
<td>0.114</td>
<td>0.106</td>
</tr>
<tr>
<td>6</td>
<td>0.127</td>
<td>0.126</td>
<td>0.126</td>
</tr>
</tbody>
</table>

Table 16: Downlink throughput for VoIP calls in presence of TCP flows (Mbps)
Figure 1: Example of the basic access mechanism

Figure 2: Sketch of the considered network
Figure 3: Homogeneous case - Aggregate throughput for $S_1$, $S_2$ and $E_1$ traffic profiles.

Figure 4: Heterogeneous case - Aggregate throughput for $S_1$ and $S_2$ nodes in scenario 1.
Figure 5: Heterogeneous case - Aggregate throughput for $S_1$, $S_2$ and $E_1$ nodes in scenario 2

Figure 6: Discrete Markov Chain describing the evolution of the number of backlogged nodes
Figure 7: VoIP - Throughput (AP and mobile nodes) for two voice codecs (G.711 and G.729)

Figure 8: $BP_s$ - Blocking probability VoIP calls
Figure 9: $BP_{e,u}$ - Blocking probability for elastic uplink flows

Figure 10: $BP_{e,d}$ - Blocking probability for elastic downlink flows
Figure 11: $ES_{e,u}$ - Average elastic throughput uplink

Figure 12: $ES_{e,d}$ - Average elastic throughput downlink